MEDIAPANEL

Instrukcja przekazywania zgód użytkownika

Serwisy WWW

Październik 2026

MEDIAPANEL

Spis treści

1. Wprowadzenie 3
2. Polityka cookie: cookies używane w gemiusPrism 3
3. Obsługa Consent Managament Platform 5

CMP zgodne z IAB 5

CMP timeout 5

CMP niezgodne z IAB 6

Korzystanie z danych w targetowaniu behawioralnym AdOcean 8

4. Rozwiązania niestandardowe 9
5. Wymagane uprawnienia 10

Content-Security-Policy 10

iframe sandbox 10

1. Wprowadzenie

Na rynkach zrzeszonych w Unii Europejskiej możesz być zobowiązany do jasnego informowania użytkowników o celach przechowywania ich danych i **zbierania potrzebnych zgód**. Istnieje kilka sposobów, aby zapytać i uszanować decyzje użytkowników w tej sprawie, a skrypt Gemius ma za zadanie je wspierać. Najczęściej używane je CMP zgodne z IAB (TCF v.1 lub 2) i dostosowanie skryptu Gemius do odczytywania wartości zgód poprzez dodanie odpowiedniego parametru. Jeśli CMP, z którego korzystasz, nie jest zgodne z IAB, nadal możesz przekazać zgody do skryptu Gemius w formacie zgodnym z IAB. Jeśli wykorzystujesz dane do targetowania behawioralnego AdOcean, postępuj zgodnie z dedykowaną instrukcją. Rozdział „Rozwiązanie niestandardowe" opisuje, w jaki sposób przekazać informacje o zgodach, jeśli zapytasz użytkowników w inny sposób niż przez CMP.

Uwaga: Gemius potrzebuje informacji **zarówno w przypadku nieudzielenia, jak i udzielenia zgody** przez użytkownika. Przypadki, w których użytkownik wyraził zgodę, a parametry przekazujące tę informację zostaną pominięte, będą traktowane jako zgoda nigdy nie wyrażona i mogą kwestionować możliwość przetwarzania przez Gemius zebranych danych.

2. Polityka cookie: cookies używane w gemiusPrism

Biorąc pod uwagę złożoność jurysdykcji, przepisów i regulacji dotyczących prywatności oraz sposobu, w jaki Ty i Twój zespół zamierzacie korzystać z narzędzi Gemius, najlepszym wyjściem przy ustalaniu, które pliki cookie muszą być wymienione na witrynie (na przykład w Twojej polityce plików cookie), jest zasięgnięcie porady doradcy prawnego. Twój radca prawny pomoże Ci określić, jak sklasyfikować wymienione poniżej pliki cookie Gemius (tj. czy są one niezbędne, analityczne, czy innego rodzaju). Nie możemy udzielić odpowiedzi na żadne pytania, które mogą mieć konsekwencje prawne.

Poniższe pliki cookie mają zastosowanie, gdy wczytany jest kod gemiusPrism:

Nazwa cookie	Opis	Typ cookie	Czas życia
Gdyn	Pozwala łączyć informacje o aktywnościach jednego użytkownika na wszystkich domenach. Dostarcza informacji wykorzystywanych przy kontroli emisji ankiet.	Third party cookie	395 dni
Gtest	Sprawdza, czy przeglądarka użytkownika akceptuje 3rd party cookie.	Third party cookie	max. 5 lat
OPTOUT	Odpowiada za anonimizację danych wykorzystywanych w badaniu Gemius. Zapobiega dalszemu wyświetlaniu ankiet.	Third party cookie	5 lat

GDPRFORGETTS	Zawiera informacje o użytkownikach korzystających z prawa do bycia zapomnianym. Odpowiada za usuwanie starych identyfikatorów użytkownika.	Third party cookie	5 lat
GNSS	Zapisuje informację, jeśli przeglądarka nie obsługuje atrybutu SameSite=None. Nadawane tylko w przeglądarkach opartych na Chromium 51-66.	Third party cookie	5 lat
__gfp_s_64b, __gfp_64b	Unikalne ID użytkownika. Pozwala na łączenie informacji o aktywnościach jednego użytkownika w obrębie domeny.	First party cookie	395 dni
__gfp_cap, __gfp_s_cap	Dostarcza informacji wykorzystywanych przy kontroli emisji ankiet Gemius first party cookie.	First party cookie	395 dni
__gfp_cache, __gfp_s_cache	Przechowuje informację na temat identyfikatora 1st party, optymalizując przydzielanie identyfikatorów w obrębie domeny.	używane jako first party cookie w Twojej domenie	395 dni

Więcej informacji na temat plików cookie można znaleźć w pliku JSON zawierającym informacje wymagane od Dostawców podczas rejestracji na globalnej liście Global Vendor List (GVL). Plik JSON jest tworzony zgodnie ze Specyfikacją techniczną TCF.

Uwaga: Plik zawiera specyfikację **wszystkich** plików cookie Gemius. Cookies związane ze skryptem gemiusPrism (pomiar gemiusAudience) zostały wymienione w wyżej zamieszczonej tabeli!

3. Obsługa Consent Managament Platform

CMP zgodne z IAB

CMP to rozwiązanie implementowane na stronie wydawcy, służące do zarządzania zgodą użytkownika na przetwarzanie danych osobowych i pozwalające na przesyłanie informacji pomiędzy dostawcami technologii. Lista zarejestrowanych przez IAB Europe dostawców CMP (Consent Management Providers, CMPs) dostępna jest na stronie https://iabeurope.eu/cmp-list/.
Aby zapewnić obsługę CMP zgodnych z IAB Europe's GDPR Transparency & Consent Framework, udostępniamy dedykowane parametry służące do przekazywania do naszych serwerów informacji na temat udzielonej zgody lub jej braku.
Uwaga: Korzystanie z parametrów nie jest obligatoryjne.
W przypadku korzystania z CMP zgodnego z IAB Framework, należy ustawić zmienną **pp_gemius_use_cmp** na **true**. Informacje na temat zgody lub jej braku zbierane są przez CMP. Jeśli użytkownik nie wyraża zgody na przetwarzanie danych przez Gemius, nie będą do Gemius wysyłane informacje na temat cookieID ani browserID. Jeśli na stronie nie zostało zamieszczone żadne CMP, można użyć zmiennej **pp_gemius_gdpr_consent**.
Uwaga: Nasz skrypt przeszukuje strukturę DOM (w górę) w celu znalezienia ramki __tcfapiLocator. Ramka ta odpowiada za przekazanie informacji na temat zgód użytkownika. Prosimy o upewnienie się, że w kodzie HTML strony ta ramka znajduje się przed skryptem Gemius.

```
<script type="text/javascript">
<!--//--><![CDATA[//><!--
var pp_gemius_identifier = 'IDENTYFIKATOR';
var pp_gemius_use_cmp = true;

// lines below shouldn't be edited
function gemius_pending(i) { window[i] = window[i] || function() {var x = window[i+'_pdata']
=   window[i+'_pdata']   ||   [];x[x.length]=arguments;};};  gemius_pending('gemius_hit');
gemius_pending('gemius_event');                              gemius_pending('gemius_init');
gemius_pending('pp_gemius_hit');                          gemius_pending('pp_gemius_event');
gemius_pending('pp_gemius_init');            (function(d,t)            {try            {var
gt=d.createElement(t),s=d.getElementsByTagName(t)[0],l='http'+((location.protocol=='https:
')?'s':'');             gt.setAttribute('async','async');gt.setAttribute('defer','defer');
gt.src=l+'://PREFIX.hit.gemius.pl/xgemius.js'; s.parentNode.insertBefore(gt,s);} catch (e)
{}})(document,'script');
//--><!]]>
</script>
```

Uwaga: W przypadku korzystania z CMP zgodnego z TCF IAB należy upewnić się, że interfejs __tcfapi jest dostępny przed uruchomieniem skryptu Gemius. Jeżeli dostawca CMP udostępnia mechanizm TCF Stub, powinien on być w pełni synchroniczny oraz zostać załadowany przed skryptem Gemius i przed innymi rozwiązaniami korzystającymi z informacji o zgodach użytkownika. Dzięki temu minimalizowane jest ryzyko problemów wynikających z kolejności ładowania zasobów i inicjalizacji CMP.

Przykład (OneTrust): W przypadku korzystania z OneTrust zalecane jest załadowanie skryptu TCF Stub https://cdn.cookielaw.org/consent/tcf.stub.js przed standardowym skryptem otSDKStub.js. Sam otSDKStub.js może pobierać dodatkowe zasoby przed udostępnieniem interfejsu TCF, co może prowadzić do sytuacji, w której skrypt Gemius uruchomi się zanim obiekt __tcfapi będzie dostępny. Dzięki zastosowaniu https://cdn.cookielaw.org/consent/tcf.stub.js interfejs TCF jest dostępny od początku ładowania strony, co pozwala skryptowi Gemius poprawnie odczytać informacje o zgodach użytkownika.

CMP timeout

Domyślnie gemiusPrism czeka na informację od CMP na temat zgody użytkownika przez 10 sekund. Aby zmienić czas oczekiwania, można użyć opcjonalnej zmiennej **pp_gemius_cmp_timeout** i przekazać w niej liczbę milisekund, przez jaką skrypt ma oczekiwać na odpowiedź CMP:

```
<script type="text/javascript">
<!--//--><![CDATA[//><!--
var pp_gemius_identifier = 'IDENTYFIKATOR';
var pp_gemius_use_cmp = true;
var pp_gemius_cmp_timeout = 10000;

// lines below shouldn't be edited
function gemius_pending(i) { window[i] = window[i] || function() {var x = window[i+'_pdata']
=   window[i+'_pdata']   ||   [];x[x.length]=arguments;};};  gemius_pending('gemius_hit');
gemius_pending('gemius_event');                                gemius_pending('gemius_init');
gemius_pending('pp_gemius_hit');                           gemius_pending('pp_gemius_event');
gemius_pending('pp_gemius_init');            (function(d,t)            {try            {var
gt=d.createElement(t),s=d.getElementsByTagName(t)[0],l='http'+((location.protocol=='https:'
)?'s':'');              gt.setAttribute('async','async');gt.setAttribute('defer','defer');
gt.src=l+'://PREFIX.hit.gemius.pl/xgemius.js'; s.parentNode.insertBefore(gt,s);} catch (e)
{}})(document,'script');
//--><!]]>
</script>
```

W przypadku opuszczenia strony przez użytkownika przed upływem zdefiniowanego czasu (**pp_gemius_cmp_timeout**) lub w przypadku braku odpowiedzi, skrypt wyśle wstrzymane hity zakładając, że zgody nie zostały udzielone.
Jeśli skrypt ma oczekiwać na odpowiedź CMP do skutku, należy ustawić wartość zmiennej **pp_gemius_cmp_timeout** na **Infinity**:

```
var pp_gemius_cmp_timeout = Infinity;
```

W efekcie, jeśli użytkownik opuści stronę przed interakcją z CMP, wstrzymane hity nie zostaną wysłane.

CMP niezgodne z IAB

W przypadku korzystania z CMP niezgodnego z IAB Framework lub w przypadku chęci nadpisania ustawień CMP, można użyć parametrów **pp_gemius_gdpr_consent** oraz **pp_gemius_gdpr_applies**.
Aby zastosować informacje na temat zgody użytkownika lub jej braku w formacie zgodnym ze standardem IAB na stronie, która nie korzysta z CMP, należy użyć zmiennej **pp_gemius_gdpr_consent = <CONSENT_STRING>;**
na przykład: **var pp_gemius_gdpr_consent ='BORViZzORViZzABABBENBK8AAAAceADAACABgApA'**; .
Uwaga: Zmienna **pp_gemius_use_cmp** ma wyższy priorytet. Jeśli **pp_gemius_use_cmp** jest ustawiona na **true** i CMP jest zamieszczone na stronie, użyte zostaną wartości podane przez CMP.

```
<script type="text/javascript">
<!--//--><![CDATA[//><!--
var pp_gemius_identifier = 'IDENTYFIKATOR';
var pp_gemius_use_cmp = false; //can be omitted
var pp_gemius_gdpr_consent = <CONSENT_STRING>;

// lines below shouldn't be edited
function gemius_pending(i) { window[i] = window[i] || function() {var x = window[i+'_pdata']
=   window[i+'_pdata']   ||   [];x[x.length]=arguments;};};  gemius_pending('gemius_hit');
gemius_pending('gemius_event');                                gemius_pending('gemius_init');
gemius_pending('pp_gemius_hit');                           gemius_pending('pp_gemius_event');
gemius_pending('pp_gemius_init');            (function(d,t)            {try            {var
gt=d.createElement(t),s=d.getElementsByTagName(t)[0],l='http'+((location.protocol=='https:
')?'s':'');             gt.setAttribute('async','async');gt.setAttribute('defer','defer');
gt.src=l+'://PREFIX.hit.gemius.pl/xgemius.js'; s.parentNode.insertBefore(gt,s);} catch (e)
{}})(document,'script');
```

Aby zdecydować, czy zapisy EU General Data Protection Regulation (GDPR, RODO) mają zastosowanie do użytkownika, należy użyć zmiennej **pp_gemius_gdpr_applies** i ustawić jej wartość na **true** lub **false**.
Jeśli RODO nie ma zastosowania w przypadku danego użytkownika, można ustawić zmienną **pp_gemius_gdpr_applies** na **false** - nie jest wtedy sprawdzana **pp_gemius_gdpr_consent** zakładając, że użytkownik wyraził zgodę. W przeciwnym razie określenie zgody lub jej braku zależy od wartości zmiennej **pp_gemius_gdpr_consent**.

```
<script type="text/javascript">
<!--//--><![CDATA[//><!--
var pp_gemius_identifier = 'IDENTYFIKATOR';
var pp_gemius_use_cmp = false; // can be omitted
var pp_gemius_gdpr_consent = <CONSENT_STRING>; //can be omitted
var pp_gemius_gdpr_applies = false;

// lines below shouldn't be edited
function gemius_pending(i) { window[i] = window[i] || function() {var x = window[i+'_pdata']
=   window[i+'_pdata']   ||   [];x[x.length]=arguments;};};  gemius_pending('gemius_hit');
gemius_pending('gemius_event');                                gemius_pending('gemius_init');
gemius_pending('pp_gemius_hit');                           gemius_pending('pp_gemius_event');
gemius_pending('pp_gemius_init');            (function(d,t)            {try            {var
gt=d.createElement(t),s=d.getElementsByTagName(t)[0],l='http'+((location.protocol=='https:
')?'s':'');             gt.setAttribute('async','async');gt.setAttribute('defer','defer');
gt.src=l+'://PREFIX.hit.gemius.pl/xgemius.js'; s.parentNode.insertBefore(gt,s);} catch (e)
{}})(document,'script');
```

Jeśli potrzebne jest opóźnienie inicjalizacji skryptu do czasu uzyskania informacji na temat zgody użytkownika, należy skorzystać z funkcji **pp_gemius_init** i zmiennej **pp_gemius_init_timeout**, opisanych w rozdziale 7.
Przykładowa definicja funkcji **pp_gemius_init**:

```
pp_gemius_init({
 'identifier': 'IDENTIFIER',
 'gdpr_consent' = <CONSENT_STRING>,
 'extraparameters' = ['language=EN', 'section_id=news'];
});
```

Korzystanie z danych w targetowaniu behawioralnym AdOcean

W przypadku korzystania z danych zebranych w gemiusPrism w targetowaniu behawioralnym AdOcean, należy użyć zmiennej **pp_gemius_dmp_purpose** i ustawić jej wartość na **true**.

```
<script type="text/javascript">
<!--//--><![CDATA[//><!--
var pp_gemius_identifier = 'IDENTYFIKATOR';
var pp_gemius_dmp_purpose = true;

// lines below shouldn't be edited
function gemius_pending(i) { window[i] = window[i] || function() {var x = window[i+'_pdata']
=   window[i+'_pdata']   ||   [];x[x.length]=arguments;};};  gemius_pending('gemius_hit');
gemius_pending('gemius_event');                                gemius_pending('gemius_init');
gemius_pending('pp_gemius_hit');                          gemius_pending('pp_gemius_event');
gemius_pending('pp_gemius_init');            (function(d,t)            {try            {var
gt=d.createElement(t),s=d.getElementsByTagName(t)[0],l='http'+((location.protocol=='https:
')?'s':'');             gt.setAttribute('async','async');gt.setAttribute('defer','defer');
gt.src=l+'://PREFIX.hit.gemius.pl/xgemius.js'; s.parentNode.insertBefore(gt,s);} catch (e)
{}})(document,'script');
//--><!]]>
</script>
```

4. Rozwiązania niestandardowe

Jeśli Twoja witryna nie korzysta z CMP zgodnego z IAB ani parametru wymienionego wyżej, właściwym sposobem uszanowania decyzji użytkownika o udostępnieniu jego danych jest użycie parametru **pp_gemius_consent**. Aby poprawnie przechowywać informacje o wyrażeniu zgody przez użytkownika, należy podać wartość parametru **pp_gemius_consent** zarówno w przypadku, gdy zgoda została udzielona, jak i gdy jej nie udzielono.
Jeśli odwiedzający wyrazi zgodę na przetwarzanie jego danych w Twojej witrynie, jesteś zobowiązany do zmiany skryptu poprzez dodanie zmiennej **pp_gemius_consent** z wartością **true**.
W sytuacji, gdy odwiedzający sprzeciwia się przetwarzaniu jego danych, należy dodać ten sam parametr **pp_gemius_consent** z wartością **false**. Należy pamiętać, aby zmienną o wartości „**false**" definiować jedynie w przypadku użytkowników, którzy nie chcą być śledzeni. Zdefiniowanie tej zmiennej za każdym razem spowoduje, że cały ruch na stronie będzie bezcookiesowy.

```
<script type="text/javascript">
<!--//--><![CDATA[//><!--
var pp_gemius_identifier = 'IDENTIFIER';
var pp_gemius_consent = false;

// lines below shouldn't be edited
function gemius_pending(i) { window[i] = window[i] || function() {var x = window[i+'_pdata']
=   window[i+'_pdata']   ||   [];x[x.length]=arguments;};};  gemius_pending('gemius_hit');
gemius_pending('gemius_event');                              gemius_pending('gemius_init');
gemius_pending('pp_gemius_hit');                         gemius_pending('pp_gemius_event');
gemius_pending('pp_gemius_init');           (function(d,t)           {try           {var
gt=d.createElement(t),s=d.getElementsByTagName(t)[0],l='http'+((location.protocol=='https:
')?'s':'');             gt.setAttribute('async','async');gt.setAttribute('defer','defer');
gt.src=l+'://PREFIX.hit.gemius.pl/xgemius.js'; s.parentNode.insertBefore(gt,s);} catch (e)
{}})(document,'script');
//--><!]]>
</script>
```

Jeśli potrzebne jest opóźnienie inicjalizacji skryptu do czasu uzyskania informacji na temat zgody użytkownika, należy skorzystać z funkcji **pp_gemius_init** i zmiennej **pp_gemius_init_timeout**, opisanych w rozdziale 7.
Przykładowa definicja funkcji **pp_gemius_init**:

```
pp_gemius_init({
  'identifier': 'IDENTIFIER',
  'consent': true,
  'extraparameters' = ['language=EN', 'section_id=news'];
});
```

5. Wymagane uprawnienia

Do poprawnego działania skryptów pomiarowych niezbędne są pewne uprawnienia, które można ograniczyć dla strony lub ramki, w której osadzony jest skrypt. Uprawnienia te można skonfigurować za pomocą nagłówka **HTTP Content-Security-Policy** lub atrybutu **sandbox** znacznika iframe.

Domyślnie wszystkie uprawnienia są nadane, więc skrypty pomiarowe będą działać. Jeśli jednak zostaną zastosowane ograniczenia, niektóre funkcje skryptu mogą przestać działać lub cały skrypt może zostać zablokowany.

Content-Security-Policy

Content-Security-Policy może zostać ustawione poprzez nagłówek HTTP lub tag <meta http-equiv> w treści strony. Dla skryptów pomiarowych wymagane są następujące ustawienia:

```
Content-Security-Policy: default-src 'self' *.hit.gemius.pl;
```

Dodatkowo, jeśli skrypty są hostowane na dedykowanej domenie wydawcy, należy uwzględnić także tę domenę, na przykład:

```
Content-Security-Policy: default-src 'self' *.hit.gemius.pl example.domain.com;
```

Jeśli używane są inne dyrektywy „**src**”, dla których **default-src** jest wartością domyślną, należy te same ustawienia zdefiniować także dla nich. W szczególności wymóg ten dotyczy:

- child-src
- connect-src
- frame-src
- img-src
- script-src

iframe sandbox

Uprawnienia skryptów w ramce można ograniczyć za pomocą atrybutu **sandbox** w tagu **<iframe>** lub nagłówka **Content-Security-Policy**. Jeśli używany jest atrybut sandbox, skrypt pomiarowy wymaga następujących ustawień:

- allow-scripts
- allow-same-origin
- allow-popups (tylko na potrzeby emisji ankiet)
- allow-popups-to-escape-sandbox (tylko na potrzeby emisji ankiet)

Przykład użycia atrybutu **sandbox**:

```
<iframe src="https://www.example.com" sandbox="allow-scripts allow-same-origin
allow-popups allow-popups-to-escape-sandbox"></iframe>
```

Permissions-Policy

Nie wolno modyfikować dyrektywy *ch-ua-high-entropy-values* w sekcji *Permissions-Policy* na stronach, na których używane są skrypty Gemius. Ograniczenie tej dyrektywy może zablokować wysyłanie Client Hints przez przeglądarkę i doprowadzić do jej **nieprawidłowej klasyfikacji.**

Poniższa konfiguracja **nie jest zalecana:**

```
Permissions-Policy: ch-ua-high-entropy-values=("self" "<PROTO>://<PREFIX>.hit.gemius.pl")
```

Może ona powodować problemy, gdy na stronie znajduje się wiele skryptów Gemius z różnymi prefiksami (np. AdOcean, DirectEffect, Heatmap). Ponieważ symbole wieloznaczne (wlidcards) są niedozwolone, konieczne byłoby stałe utrzymywanie pełnej i poprawnej listy prefiksów, a takie rozwiązanie jest podatne na błędy.

Gemius S.A.
ul. Domaniewska 48
02-672 Warszawa, Polska

Kontakt:
mediapanel@gemius.com
+ 48 22 390 90 90
+ 48 22 378 30 50